Commercial Solutions for Classified (CSfC) Selections for Transport Layer Security (TLS) Software Applications

Overview

Transport Layer Security (TLS) software application products (i.e., TLS Client as defined in the Mobile Access (MA) Capability Package (CP)) used in Commercial Solutions for Classified (CSfC) solutions shall be validated by National Information Assurance Partnership (NIAP)/Common Criteria Evaluation and Validation Scheme (CCEVS) or Common Criteria Recognition Arrangement (CCRA) partnering schemes as complying with the current requirements of NIAP's Protection Profile for Application Software Version 1.3 (pp_app_v1.3) and Functional Package for Transport Layer Security (TLS) v1.1 (pkg_tls_v1.1). This validated compliance shall include the selectable requirements contained in this document.

TLS can be used/implemented in many different ways, threats and technology continuously progress, and TLS continues to evolve, which may cause the below selections to change or become obsolete. The objective of the below selections is to provide information to enable the use of the Commercial National Security Algorithm Suite (CNSA Suite) and support the use of TLS software applications in CSfC Solutions.

Please provide questions, comments on usability, applicability, and/or shortcomings to the CSfC Program (csfc@nsa.gov).

Notes

Note 1: CSfC TLS software applications must be configured to support (i.e., implement the functionality or invoke platform-provided functionality) and have documented configurations in the Security Target (ST) and Admin Guide for the functionality described in the Selections for the below Security Functional Requirements (SFRs):

- FCS_HTTPS_EXT.1/Client (if the Target of Evaluation (TOE) uses HTTPS)
- FIA_X509_EXT.1
- FIA_X509_EXT.2
- Functional Package for Transport Layer Security (TLS) v1.1 Selections

Even if the SFRs aren't required by the PP due to specific selections made by the ST author, the functionality is required to enable TLS software applications to comply with CSfC Capability Package (CP) requirements (e.g., CNSA Suite, TLS capabilities). For example, if the TLS software application invokes platform-provided functionality in FTP_DIT_EXT.1.1, then per PP design, the SFRs listed above may not be evaluated, but CSfC requires supported and documented functionality for the SFRs (i.e., demonstrate equivalent functionality as noted above) as part of CSfC Components List product eligibility.

Note 2: The following selections apply to CSfC TLS software application functionality. If needed, functionality and/or configurations outside the scope of a CSfC TLS software application that conflict with the CSfC selections could be NIAP validated using a separate iteration of the SFR. The ST author should document that the iteration of the SFR shouldn't be used to validate compliance with CSfC

selections and the configuration isn't part of the NIAP-certified evaluated configuration for CSfC TLS software application Use Cases.

Document Conventions

The conventions used in descriptions of the document are as follows:

- Assignment completed within a selection in the PP: the completed assignment text is indicated with ***italicized and underlined text*** (i.e., CSfC mandatory completed assignments/selections unless otherwise indicated by the text "at least one of the following underlined selections")
- Assignment partially completed in the PP: indicated with *italicized text*
- Refinement text is indicated with ~~strikethroughs~~
- Additional clarifying text or CSfC specific language is indicated with `light blue Courier New Text`
- Links to sources, additional information, and email addresses are indicated with blue underlined text.

Protection Profile for Application Software Version 1.3 Selections

FCS_RBG_EXT.1.1

The application shall `perform at least one of the following` [**Selection:**

- ~~*use no DRBG functionality*~~,
- *invoke platform-provided DRBG functionality*,
- *implement DRBG functionality*

] for its cryptographic operations.

FCS_CKM_EXT.1.1

The application shall `perform at least one of the following` [**Selection:**

- ~~*generate no asymmetric cryptographic keys*~~,
- *invoke platform-provided functionality for asymmetric key generation*,
- *implement asymmetric key generation*

].

FCS_RBG_EXT.2.1

`If the application performs DRBG services` the application shall perform all deterministic random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using `at least one of the following` [**Selection:** *Hash_DRBG (`SHA-384`, `SHA-512`), HMAC_DRBG (`SHA-384`, `SHA-512`), CTR_DRBG (`AES-256`)*].

Application Note: The objective of the CSfC specific language for DRBG algorithms is to ensure compatibility with the CSfC CPs by selecting compliant algorithms that provide the required security strength.

FCS_RBG_EXT.2.2

If the application performs DRBG services the deterministic RBG shall be seeded by an entropy source that accumulates entropy from a platform-based DRBG and [**Selection**:

- *a software-based noise source*,
- *a hardware-based noise source*,
- *no other noise source*

] with a minimum of [Selection:

- ~~*128 bits*~~
- *256 bits*

] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

FCS_CKM.1.1(1)

The application shall perform at least one of the following [**Selection:**

- *invoke platform-provided functionality for asymmetric key generation,*
- *implement asymmetric key generation*

] to generate asymmetric cryptographic keys in accordance with at least one of the following specified cryptographic key generation algorithms [**Selection:**

- ***[RSA schemes]*** *using cryptographic key sizes of [2048-bit and 3072-bits or greater] that meet the following FIPS PUB 186-4, "Digital Signature Standard (DSS), Appendix B.3"*
- ***[ECC schemes]*** *using NIST curves P-256, P-384 and* [*Selection*: ~~*P-521*~~, *no other curves*] *that meet the following: FIPS PUB 186-4, "Digital Signature Standard(DSS)", Appendix B.4*,
- ~~*[**FFC schemes**] using cryptographic key sizes of [2048-bit or greater] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1]*~~,
- ~~*[FFC Schemes] **using Diffie-Hellman group 14** that meet the following: **RFC 3526, Section 3***~~,
- ***[FFC Schemes]*** *using "safe-prime" groups that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and* [***Selection***: *RFC 3526, RFC 7919*]

].

FCS_CKM.1.1(2)

If symmetric keys are generated, the application shall generate symmetric cryptographic keys using a Random Bit Generator as specified in FCS_RBG_EXT.1 and specified cryptographic key sizes [Selection:

- ~~*128 bit*~~,
- *256 bit*

].

FCS_COP.1.1(1)
If the applications implements cryptographic functionality, the **application** shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm [**Selection**:

- *AES-CBC (as defined in NIST SP 800-38A) mode*,
- *AES-GCM (as defined in NIST SP 800-38D) mode*,
- *AES-XTS (as defined in NIST SP 800-38E) mode*
- *AES-CCM (as defined in NIST SP 800-38C) mode (See TD0540)*

] and cryptographic key sizes [Selection: ~~*128-bit*~~, *256-bit*].

FCS_STO_EXT.1.1
The application shall perform at least one of the following [**Selection:**

- *not store any credentials*,
- *invoke the functionality provided by the platform to securely store [**Assignment**: applicable secret keys, PKI private keys, and/or passwords*],
- *implement functionality to securely store [**Assignment**: applicable secret keys, PKI private keys, and/or passwords*] *according to* [***Selection*:** *FCS_COP.1(1), FCS_CKM.1(3)*]

] to non-volatile memory.

FTP_DIT_EXT.1.1
The application shall perform at least one of the following [**Selection**:

- ~~*not transmit any* [selection: *data, sensitive data*]~~,
- *encrypt all transmitted* [selection: ~~*sensitive data*~~, *data*] with at least one of the following underlined selections [**Selection**: *HTTPS in accordance with FCS_HTTPS_EXT.1/Client, TLS as defined in the TLS Package, ~~DTLS as defined in the TLS Package~~, SSH as conforming to the Extended Package for Secure Shell, IPsec as defined in the PP-Module for VPN Client*],
- ~~*invoke platform-provided functionality to encrypt all transmitted sensitive data with* [selection: *HTTPS, TLS, DTLS, SSH*]~~,
- *invoke platform-provided functionality to encrypt all transmitted data with* at least one of the following underlined selections [**Selection**: *HTTPS, TLS, ~~DTLS~~, SSH*]

] between itself and another trusted IT product (TLS Protected Servers). (See TD0473)

FCS_HTTPS_EXT.1.1/Client
If the TOE uses HTTPS, the application shall implement the HTTPS protocol that complies with RFC 2818. (See TD0473)

FCS_HTTPS_EXT.1.2/Client
If the TOE uses HTTPS, the application shall implement HTTPS using TLS as defined in the TLS package. (See TD0473)

FCS_HTTPS_EXT.1.3/Client
If the TOE uses HTTPS, the application shall perform at least one of the following [**Selection:** ~~*not establish the application-initiated connection*~~, *notify the user and not establish the user-initiated connection, notify the user and request authorization to establish the user-initiated connection*] if the peer certificate is deemed invalid. (See TD0473)

FCS_CKM.2.1

The application shall perform at least one of the following [**Selection:** *invoke platform-provided functionality*, *implement functionality*] to perform cryptographic key establishment in accordance with at least one of the following specified cryptographic key establishment methods

[**Selection:**

- ~~***[RSA-based key establishment schemes]*** *that meets the following:* ***[NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography"] ,***~~
- ~~***[RSA-based key establishment schemes]*** *that meet the following:* ***RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1"***~~
- ***[Elliptic curve-based key establishment schemes]*** *that meets the following:* ***[NIST Special Publication 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography]***],
- ~~***[Finite field-based key establishment schemes]*** *that meets the following:* ***[NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"] ,***~~
- ~~***[Key establishment scheme using Diffie-Hellman group 14]*** *that meets the following:* ***RFC 3526, Section 3***~~,
- ***[FFC Schemes using "safe-prime" groups]*** *that meet the following: ['****NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography****"] and* [**Selection**: *RFC 3526, RFC 7919*]]

].

FCS_COP.1.1(2)

The **application** shall perform *cryptographic hashing* services in accordance with a specified cryptographic algorithm [**Selection:**

- ~~*SHA-1,*~~
- *SHA-256,*
- *SHA-384,*
- *SHA-512*
- ~~*no other*~~

] and message digest sizes [**Selection:**

- ~~*160,*~~
- *256,*
- *384,*
- *512*
- ~~*no other*~~

] bits that meet the following: FIPS Pub 180-4.

FCS_COP.1.1(3)
The **application** shall perform *cryptographic signature services (generation and verification)* in accordance with at least one of the following specified cryptographic algorithms [**Selection**:

- ***RSA schemes*** *using cryptographic key sizes of 2048-bit* and 3072-bits or *greater that meet the following: FIPS PUB 186-4, Digital Signature Standard (DSS), Section 4,*
- ***ECDSA schemes*** *using NIST curves, P-256, P-384 and [Selection:~~P-521~~, no other curves] that meet the following: FIPS PUB 186-4, Digital Signature Standard (DSS), Section 5*

].

FCS_COP.1.1(4)
The **application** shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm

- HMAC-SHA-256

and [**Selection**:

- ~~*SHA-1*~~,
- *SHA-384*,
- *SHA-512*
- ~~*no other algorithms*~~

] with key sizes [**Assignment**: *key size(s) in bits ≥ the message digest size(s)*] and message digest sizes 256 and [**Selection**: ~~*160*~~, *384*, *512*, ~~*no other sizes*~~] bits that meet the following: FIPS Pub 198-1 *The Keyed-Hash Message Authentication Code* and FIPS Pub 180-4 *Secure Hash Standard*.

FIA_X509_EXT.1.1
The application shall perform at least one of the following [**Selection:** *invoke platform-provided functionality*, *implement functionality*] to validate certificates in accordance with the following rules: (see TD0587)

- RFC 5280 certificate validation and certificate path validation
- The certificate path must terminate with a trusted CA certificate
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met
- The application shall validate that any CA certificate includes caSigning purpose in the key usage field
- The application shall validate the revocation status of the certificate using at least one of the following [**Selection:** *the Online Certificate Status Protocol (OCSP) as specified in ~~RFC 2560~~* RFC 6960 (see TD0587)*, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3* and *a Certificate Revocation List (CRL) as specified in ~~RFC 5759~~* RFC 8603, *~~an OCSP TLS Status Request Extension (i.e., OCSP stapling) as specified in RFC 6066~~*, *~~OCSP TLS Multi-Certificate Status Request Extension (i.e., OCSP Multi-Stapling) as specified in RFC 6961~~*].
- The application shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the extendedKeyUsage field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
    - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the extendedKeyUsage field.

FIA_X509_EXT.2.1
The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for at least one of the following underlined selections [**Selection**: *HTTPS*, *TLS*, *~~DTLS~~*, *SSH*, *IPsec*].

FIA_X509_EXT.2.2
When the application cannot establish a connection to determine the validity of a certificate, the application shall perform at least one of the following [**Selection:** *allow the administrator to choose whether to accept the certificate in these cases*, *~~accept the certificate~~*, *not accept the certificate*].

Functional Package for Transport Layer Security (TLS) v1.1 Selections

FCS_TLS_EXT.1.1
The product shall implement [Selection:

- *TLS as a client*

].

FCS_TLSC_EXT.1.1
The product shall implement TLS 1.2 (RFC 5246) and [Selection: *no earlier TLS versions*] as a client that supports at least one of the following cipher suites [**Selection:**

- ~~*TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246,*~~
- ~~*TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,*~~
- ~~*TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,*~~
- ~~*TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,*~~
- ~~*TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,*~~
- ~~*TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,*~~
- *TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,*
- ~~*TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,*~~
- ~~*TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,*~~
- ~~*TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*~~,
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,*
- ~~*TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,*~~
- ~~*TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,*~~
- ~~*TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*~~,
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*

] and also supports functionality for [Selection:

- *mutual authentication*
- ~~*session renegotiation,*~~
- ~~*none*~~

].

FCS_TLSC_EXT.1.2
The product shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3
The product shall not establish a trusted channel if the server certificate is invalid [**Selection:**

- *with no exceptions* or
- *except when override is authorized*

].

FCS_TLSC_EXT.2.1
The product shall support mutual authentication using X.509v3 certificates.

FCS_TLSC_EXT.3.1
The product shall present the signature_algorithms extension in the Client Hello with the supported_signature_algorithms value containing the following hash algorithms: [**Selection**: ~~*SHA256*~~, *SHA384*, *SHA512*] and no other hash algorithms.

FCS_TLSC_EXT.5.1

The product shall present the Supported Groups Extension in the Client Hello with at least one of the following underlined supported groups [**Selection:**

- ~~*secp256r1*~~,
- *secp384r1*,
- ~~*secp521r1*~~,
- ~~*ffdhe2048(256)*~~,
- *ffdhe3072(257)*,
- *ffdhe4096(258)*,
- ~~*ffdhe6144(259)*~~,
- ~~*ffdhe8192(260)*~~

].